

REMARKS

The Examiner has rejected Claims 1, 8, 15, 22-24 and 26 under 35 U.S.C. 103(a) as being unpatentable over Moeller et al. (U.S. Patent No. 5,473,777) in view of Sitbon et al. (U.S. Patent No. 5,568,487). Applicant respectfully disagrees with this rejection.

With respect to independent Claims 1, 8, 15 and 22, the Examiner relies on the following excerpts from Moeller to make a prior art showing of applicant's claimed "selecting applications from a group of applications adapted for working in conjunction with a first application program interface to gain access to a network, the first application interface adapted for permitting the applications to gain access to the network."

"As will be appreciated by those skilled in the relevant art, the application programs 130A, 130B, and 132 cannot interact directly with the operating system 114 unless the operating system 114 implements an environment in which the application programs 130A, 130B, and 132 are adapted to operate. For example, if the application 132 is adapted to operate in the IBM PS/2 environment, then the application 132 cannot directly interact with the operating system 114 unless the operating system 114 is the IBM PS/2 operating system (or compatible)." (Col. 6, lines 9-15) (emphasis added)

"In accordance with the present invention, the object-oriented class library 402 includes related object-oriented classes for enabling an object-oriented application (such as the applications 130A and 130B) to access in an object-oriented manner services provided by the operating system 114. The object-oriented classes comprise methods which include procedural function calls compatible with the native procedural interface of the operating system 114." (Col. 6, lines 45-54) (emphasis added)

"The object-oriented class library comprises related object-oriented classes for enabling the application to access in an object-oriented manner services provided by the operating system. The object-oriented classes include methods for accessing the operating system services using procedural function calls compatible with the native procedural interface of the operating system." (Col. 3, lines 52-58) (emphasis added)

Applicant respectfully asserts that simply implementing an environment in which the application programs are adapted to operate, as disclosed in Moeller as cited above, does not meet applicant's claimed "selecting applications from a group of applications adapted for working in conjunction with a first application program interface" (emphasis added). Moeller suggests implementing an environment, such as an operating system, in which ALL application programs can operate. Applicant's claim language expressly requires for applications to be selected from a group of applications, in which each application in the group of applications is adapted to work with a first application program interface. There is simply no such selection in Moeller, as set forth in the context of the remaining claim elements.

Further, implementing "related object-oriented classes for enabling an object-oriented application...to access...object-oriented manner services provided by the operating system", as disclosed in Moeller per the citations above, does not meet "selecting applications from a group of applications adapted for working in conjunction with a first application program interface to gain access to a network," as claimed by applicant. Specifically, Moeller's teaching of utilizing object-oriented classes fails to even suggest a group of applications that are adapted for working in conjunction with a first application program interface to gain access to a network.

The Examiner also relies on the following excerpt from Moeller to make a prior art showing of applicant's claimed "second application program interface."

"accesses to procedural function calls compatible with the procedural interface of the operating system 114." (see col. 10, lines 30-31)

After carefully reviewing such excerpts, and the remaining Moeller reference, it is clear that mere procedural function calls are provided, in the context of Moeller, to create and manipulate a virtual memory range in a memory component. Creating and manipulating a virtual memory range in a memory component by way of procedural functions clearly fail to meet applicant's claimed "second application program interface adapted for precluding the applications from accessing" (emphasis added). Only applicant teaches and claims such a second

application program interface with such specific purpose, which is namely to *preclude* applications from accessing a network.

It thus appears that the Examiner has not taken into consideration the full weight of applicant's claims. Only applicant teaches two separate application program interfaces with such specific claimed purposes, wherein the first application is adapted to work with the group of applications to grant access to a network and the second application is adapted to preclude selected applications from accessing the network, in the specific context claimed.

Still yet, the Examiner relies on the following excerpts from Sitbon to make a prior art showing of applicant's claimed "installing a second application program interface adapted for precluding the applications from accessing the network; and wrapping the selected applications for allowing the selected applications to access the network via the second application program interface, where the selected applications would otherwise be precluded network access by the second application program interface."

"FIGS. 3a and 3b, for the sake of better comprehension, will be read together with FIG. 4, which shows a table of correspondence between the "socket" interface calls ("client" side) and the "XTI" interface ("server" side). When the "socket" interface calls have no correspondence with the "XTI" interface, specific interface routines are used, which are represented by the letter (R) in FIG. 4." (Col. 5, lines 50-55) (emphasis added)

"In fact, for a standard TCP/IP application of the FTP (file transfer protocol), Telnet (virtual terminal management), RPC (remote procedure call) or DCE (distributed computing environment) type to be able to be run on OSI, the source code has to be taken up and updated or modified, then all the calls have to be run through in reverse order, the primitives have to be called, and then addressing has to be redone, and because such an operation was especially tedious and complicated, it was used little if at all." (Col. 1, lines 40-49) (emphasis added)

"These various calls SC+SY are then rerouted to the wrapper W, at the moment of the link editing phase before the executable is obtained. The second object of the wrapper W is to automatically convert the addresses specific to the TCP/IP network into addresses of the OSI/CO network, and to enable the passage from the TCP/IP protocol to the OSI/CO protocol.

After conversion, the calls SC+SY intended for the TCP/IP network are transmitted to the "XTI" interface so as to be used directly in the OSI/CO interface, which is the third object of the wrapper W." (Col. 3, lines 5-15) (emphasis added)

Applicant respectfully asserts that Sitbon's disclosed "'socket' interface calls have no correspondence" and "the source code has to be taken up and updated or modified," as cited above, does not even suggest "installing a second application program interface adapted for precluding the applications from accessing the network," as claimed by applicant. Sitbon simply recognizes that there is no correspondence between a socket interface call and an "XTI" and teaches that such calls may be converted in order to be able to run on an OSI. This clearly fails to suggest installing a second application program interface that precludes applications from accessing the network.

Further, Sitbon teaches converting source code to be able to create access between a TCP/IP interface and an OSI interface, whereas applicant claims installing a second application that precludes access to a network. Thus, Sitbon *teaches away* from applicant's claimed feature.

For substantially the same reasons (but not identical to the extent that the claim language differs), applicant respectfully asserts that independent Claims 23 and 24 are also deemed novel and unobvious in view of the Moeller and Sitbon references.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *In re Vaack*, 947 F.2d 488, 20 USPQ2d 1438 (Fed.Cir.1991).

Applicant respectfully asserts that at least the first and third elements of the *prima facie* case of obviousness have not been met by the prior art. Specifically, the Moeller and Sitbon references fail to teach or suggest all of the claim limitations, and any necessary combination/modification would be unobvious.

A notice of allowance or a specific prior art showing of all of applicant's claim limitations, in combination with the remaining claim elements, is respectfully requested.

It is further noted that the Examiner's rejection is further replete with deficiencies with respect to the dependent claims. Just be way of example, with respect to Claim 4 et al., the Examiner has rejected the subject matter of such claim, as well as Claims 2-3, 5-6, 9-13, 16-20, and 25, under 35 U.S.C. 103(a) as being unpatentable over Moeller et al. (U.S. Patent No. 5,473,777) in view of Sitbon et al. (U.S. Patent No. 5,568,487), and further in view of OPT (Optimizations).

Specifically, the Examiner relies on the following Moeller excerpt to make a prior art showing of applicant's claimed "wherein the extractor code is further adapted for interfacing with the second application program interface" (see Claim 4 et al.).

"The code library 110 may represent multiple code libraries (not shown) related to the wrapper 128, wherein each of the code libraries include the computer program logic for one of the object-oriented classes of the class library 402." (see col. 9, lines 1-5)

After careful review of such excerpt, however, it is clear that it fails to even suggest extractor code, let alone extractor code that is further adapted for interfacing with the second application program interface, as claimed. The above excerpt from Moeller does not even suggest extractor code, that can extract data, and that can also interface with a second application program interface. Rather, such excerpt simply suggests code libraries that each include logic for object-oriented classes of the class library.

Even still, the Examiner relies on the following Moeller excerpt to make a prior art showing of applicant's claimed "wherein the location in memory is where a routine is stored for allowing the selected applications to access the network" (see Claim 6 et al.).

"The library server processes the request by accessing the desired computer program logic from the code library and sending the desired computer program logic to the area of memory designated by the destination address." (see col. 9, lines 17-20)

Such excerpt, however, fails to even suggest any sort of location in memory where a routine is stored for allowing the selected applications to access the network, as claimed. Applicant argues that a library server that sends logic to an area of memory simply does not meet the specificity of applicant's claims, as noted above.

Still yet, applicant brings to the Examiner's attention the following additional dependent claims which were previously incorporated, but have still not been considered:

"wherein the second application program interface includes a modified copy of the first application program interface" (see Claim 28); and

"wherein the second application program interface is separate from the first application program interface" (see Claim 29).

Again, a notice of allowance or a specific prior art showing of all of applicant's claim limitations, in combination with the remaining claim elements, is respectfully requested.

Thus, each of the independent claims is deemed allowable along with any claims depending therefrom. Reconsideration is respectfully requested.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 505-5100. If any fees are due in connection with the filing of this paper, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1351 (Order No. NAI1P096/02.015.01).

Respectfully submitted,


Kevin J. Zilka
Registration No. 41,429

P.O. Box 721120
San Jose, CA 95172
Telephone: (408) 505-5100